



[www.chameleoncloud.org](http://www.chameleoncloud.org)

## FROM TESTBED TO INSTRUMENT: EXPERIMENT SUMMARIES AND REPEATABILITY SERVICES IN CHAMELEON

**Kate Keahey**

Mathematics and CS Division, Argonne National Laboratory  
Computation Institute, University of Chicago  
*keahey@anl.gov*

*November 15, 2017*

*SCI7 BOF on Practical Reproducibility*

DECEMBER 6, 2017

I



# CHAMELEON IN A NUTSHELL

- ▶ **Open** production testbed for **Computer Science Research**
  - ▶ Available since 07/2015, just renewed for another 3 years
  - ▶ Currently 1,900+ users, 300+ projects, 100+ institutions, 100+ publications
- ▶ **Large-scale:** “Big Data, Big Compute research”
  - ▶ ~650 nodes (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network
  - ▶ Heterogeneous hardware: GPUs, FPGAs, ARMs, Atoms, and others
  - ▶ Coming very soon: IceLake nodes and Corsa switches
- ▶ **Reconfigurable:** “As close as possible to having it in your lab”
  - ▶ Deep reconfigurability (bare metal) and isolation
  - ▶ Power on/off, reboot from custom kernel, serial console access, etc.
- ▶ Blueprint for a **sustainable** production testbed: “cost-effective to deploy, operate, and enhance”
  - ▶ Powered by OpenStack with bare metal reconfiguration (Ironic)
  - ▶ Our contributions now recognized as official OpenStack component!

# TOWARDS A SCIENTIFIC INSTRUMENT



- ▶ **Deploy:** what we are doing today – and more
- ▶ **Capture:** observe, monitor, measure – easily
- ▶ **Record:** a comprehensive “active record”
  - ▶ Re-examine, share/publish, review, re-play

# CAPTURE: THE FOUNDATION

- ▶ Testbed versioning
  - ▶ Fine-grain representation
  - ▶ Automated discovery and updates
  - ▶ 53 versions since public availability – and counting
  - ▶ Still working on: better firmware version management
- ▶ Appliance management
  - ▶ Configuration, versioning, publication
  - ▶ Still working on: repository vs catalog connection
- ▶ Monitoring and logging
  - ▶ Making it accessible in easier ways
- ▶ However... the user still has to keep track of this information

# CAPTURE: KEEPING TRACK OF EXPERIMENTS

- ▶ Everything in a testbed is a recorded event
    - ▶ The resources you used
    - ▶ The appliance/image you deployed
    - ▶ The monitoring information your experiment generated
    - ▶ Plus any information you choose to share with us: e.g., “start power\_exp\_23” and “stop power\_exp\_23”
- 
- ▶ **Experiment précis:** information about your experiment made available in a “consumable” form
  - ▶ (Bonus: it can be integrated with many existing tools, e.g., Jupyter or Grafana)...

# RECORD: MOVING TOWARDS REPEATABILITY

- ▶ Experiment précis
  - ▶ Recording the experiment for you: closing the gap between resource versions, appliances, and events
  - ▶ “Active record” that can be given to a reviewer or shared with others
- ▶ Publishing experiment précis
- ▶ Integration with popular tools
- ▶ From experiment précis to experiment replays
  - ▶ Model-based experiment capture
  - ▶ Re-play tools

# PARTING THOUGHTS

- ▶ The barrier to repeatability/reproducibility is the cost
  - ▶ Lost opportunity to do other research
- ▶ Infrastructure projects are well positioned to lower this cost by recording your experiments for you
- ▶ ... but won't cover the whole workflow
- ▶ Make it cheaper to create repeatable experiments than non-repeatable experiments
  - ▶ Incentives: reviews, iterations, challenges/awards, supporting students moving through the lab, etc.
  - ▶ Tools: notebooks, visualizations, data processing, experimental workflow management, etc.